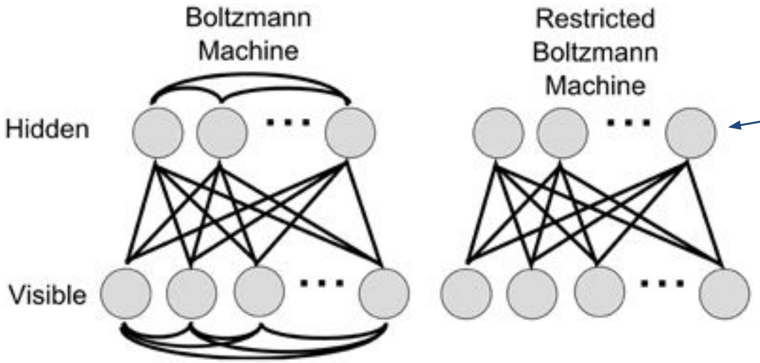
FORNAX

How do the RBMs work?

Krzysztof Kolasiński March - 2017

What is RBM

It is a Boltzmann machine with restriction



latent space

for example pixels of image

This restriction leads to significant simplification of the model

What is RBM

RBMs are defined in term of energy function

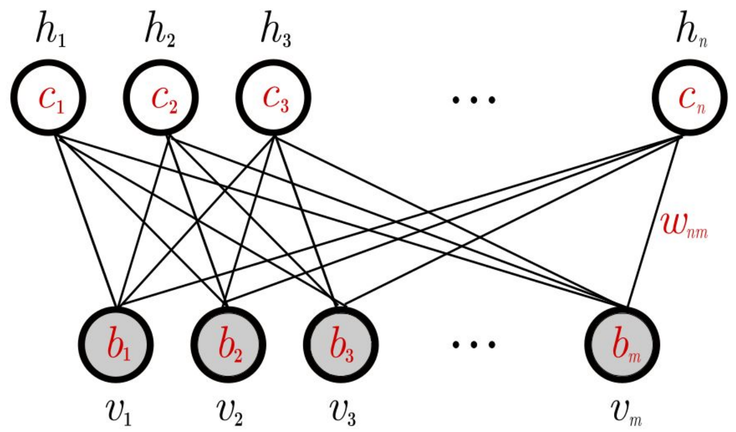


Fig. 5. The network graph of an RBM with n hidden and m visible units.

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

the random variables $(\boldsymbol{V},\boldsymbol{H})$ take values $(\boldsymbol{v},\boldsymbol{h}) \in \{0,1\}^{m+n}$

$\boldsymbol{V} = (V_1,\ldots,V_m)$ - *represent the observable data*

$\boldsymbol{H} = (H_1,\ldots,H_n)$ - *to capture the dependencies between observed variables*

W *is a matrix which connects the visible and hidden units*
b, **c** *are the bias terms associated with visible and hidden units*

What is RBM

RBMs are defined in term of energy function

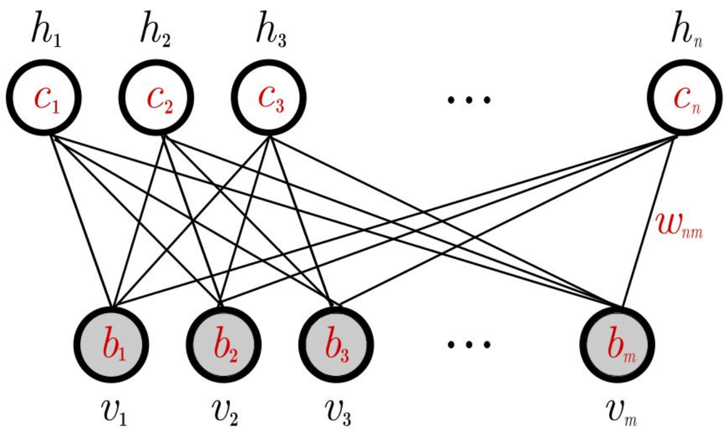


Fig. 5. The network graph of an RBM with n hidden and m visible units.

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

the random variables $(\boldsymbol{V},\boldsymbol{H})$ take values $(\boldsymbol{v},\boldsymbol{h}) \in \{0,1\}^{m+n}$

For a given configuration of units V and H we can compute energy as:

$$E \;=\; -\mathbf{h}^T\mathbf{W}\mathbf{v} - \mathbf{h}^T\mathbf{b} - \mathbf{v}^T\mathbf{c}$$

```
function rbm_energy(v, h, rbm::RBM)
  return -(v' * rbm.w * h + h' * rbm.bh + v' * rbm.bv)
end
```

What is RBM

RBMs are defined in term of energy function

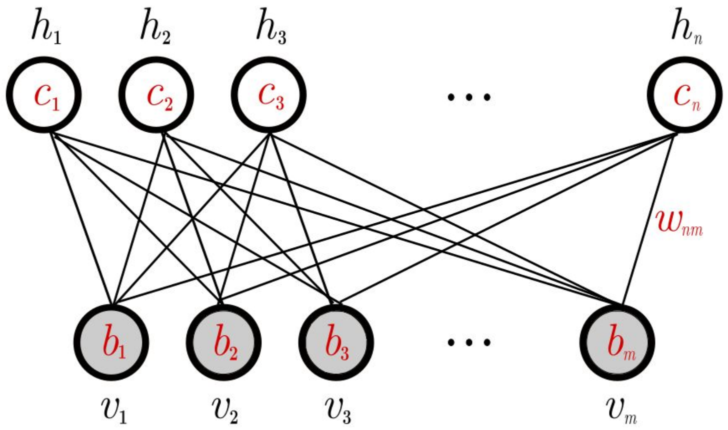


Fig. 5. The network graph of an RBM with n hidden and m visible units.

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

the random variables $(\boldsymbol{V}, \boldsymbol{H})$ take values $(\boldsymbol{v}, \boldsymbol{h}) \in \{0, 1\}^{m+n}$

For a given configuration of units V and H we can compute energy as:

$$E = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{h}^T \mathbf{b} - \mathbf{v}^T \mathbf{c}$$

Each configuration of using has defined probability (Boltzmann distribution)

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v}, \boldsymbol{h})}$$

where $Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$ is a partition function

On partition function

It contains sum of all possible hidden/visible configurations

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

Let us consider the case where **V** = $\{v_1, v_2, v_3\}$ and **H** = $\{h_1, h_2\}$.

For **V** we have 2^3 possible configurations: {0,0,0}, {0,0,1}, {0,1,0}, {0,1,1}, {1,0,0}, {1,0,1}, {1,1,0}, {1,1,1}
For **H** we have 2^2 possible configurations: {0,0}, {0,1}, {1,0}, {1,1}

In real world examples the Z is intractable, for example MNIST images has dimensions of 28x28 pixels. This gives 2^{28x28} possible configurations.

On partition function

It contains sum of all possible hidden/visible configurations

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

Let us consider the case where **V** = $\{v_1, v_2, v_3\}$ and **H** = $\{h_1, h_2\}$.

For **V** we have 2^3 possible configurations: {0,0,0}, {0,0,1}, {0,1,0}, {0,1,1}, {1,0,0}, {1,0,1}, {1,1,0}, {1,1,1}
For **H** we have 2^2 possible configurations: {0,0}, {0,1}, {1,0}, {1,1}

In the example above the sum can be computed easily:

$$Z = e^{-E(\{0,0,0\},\{0,0\})} + e^{-E(\{0,0,0\},\{0,1\})} + \ldots + e^{-E(\{1,1,1\},\{1,1\})}$$

```
vp = get_permutations(length(rbm.bv))
hp = get_permutations(length(rbm.bh))
Z = Z_exact(rbm, hp, vp)
```

```
function Z_exact(rbm::RBM, hp, vp)
  Z = 0 # partition function
  for i = 1:length(hp)
    for j = 1:length(vp)
      Z += exp(-rbm_energy(vp[j], hp[i], rbm))
    end
  end
  return Z
end
```

What do we want to maximize?

The RBM: $E = -\mathbf{h}^T\mathbf{W}\mathbf{v} - \mathbf{h}^T\mathbf{b} - \mathbf{v}^T\mathbf{c}$ $\quad Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$

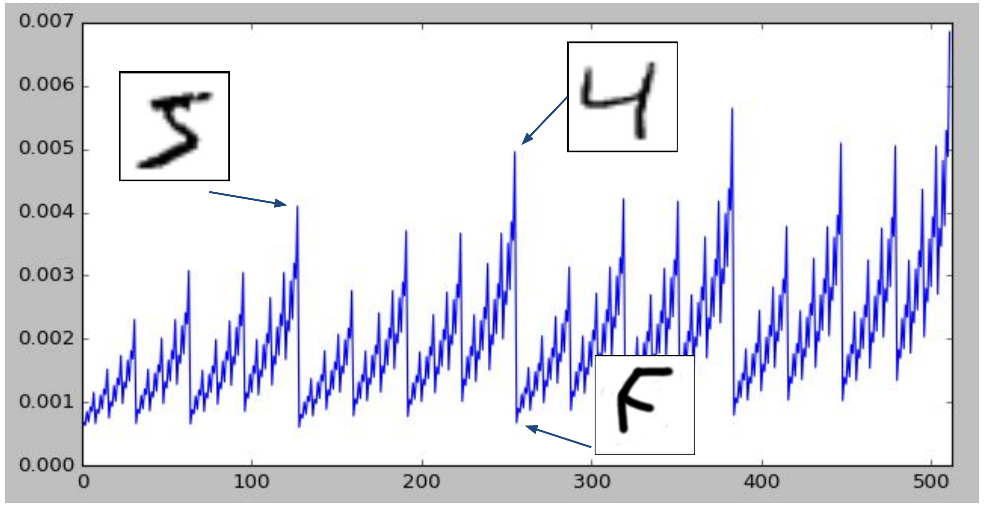
$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

We want to find **W**, **b**, **c** such that the probability of observing **V** will be maximal

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

marginal distribution of ***V***

Example of the concept: high probabilities for pictures similar as in data set



What do we want to maximize?

The RBM: $E = -\mathbf{h}^T\mathbf{W}\mathbf{v} - \mathbf{h}^T\mathbf{b} - \mathbf{v}^T\mathbf{c}$ $\quad Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$

$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

We want to find **W**, **b**, **c** such that the probability of observing **V** will be maximal

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

We look for optimal (W, b, c) = **theta** using standard **gradient ascent** on log-likelihood

$$\ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v}) = \ln p(\boldsymbol{v} \mid \boldsymbol{\theta}) = \ln \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \underbrace{\eta \frac{\partial}{\partial \boldsymbol{\theta}^{(t)}} \left(\ln \mathcal{L}(\boldsymbol{\theta}^{(t)} \mid S)\right) - \lambda \boldsymbol{\theta}^{(t)} + \nu \Delta \boldsymbol{\theta}^{(t-1)}}_{= \Delta \boldsymbol{\theta}^{(t)}}$$

Regularization + momentum

Expression for gradients

We are looking for expression of gradient of L with respect to model parameters (**theta**)

$$\ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v}) = \ln p(\boldsymbol{v} \mid \boldsymbol{\theta}) = \ln \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} = \ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} - \ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

It's a simple math:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{v})}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \right) - \frac{\partial}{\partial \boldsymbol{\theta}} \left(\ln \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \right)$$

$$= -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{v}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}} + \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

$$p(\boldsymbol{h} \mid \boldsymbol{v}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{p(\boldsymbol{v})} = \frac{\frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}$$

$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

This is still intractable

Finding expression for p(h| v)

Due to the structure of RBMs the conditional probability p(h| v) can be found analytically

$$-\sum_{\boldsymbol{h}} \boxed{p(\boldsymbol{h}\,|\,\boldsymbol{v})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

Let's group E by h_i

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i}^{n} h_i \boxed{\left(\sum_{j}^{m} \omega_{ij} v_j + c_i\right)} - \boxed{\sum_{j}^{m} b_j v_j}$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i}^{n} h_i \Delta_i - E_v$$

$$p(\boldsymbol{h}\,|\,\boldsymbol{v}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{p(\boldsymbol{v})} = \frac{\frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}$$

Let's write it explicitly

$$e^{-E(\mathbf{v},\mathbf{h})} = e^{\sum_i^n h_i \Delta_i + E_v} = e^{\sum_i^n h_i \Delta_i} e^{E_v} = \boxed{e^{E_v} \prod_{i}^{n} e^{h_i \Delta_i}}$$

$$\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} = \sum_{\mathbf{h}} \boxed{e^{E_v} \prod_{i}^{n} e^{h_i \Delta_i}} = e^{E_v} \boxed{\sum_{\mathbf{h}} \prod_{i}^{n} e^{h_i \Delta_i}}$$

$$e^{E_v} \sum_{\mathbf{h}} \prod_{i}^{n} e^{h_i \Delta_i} = e^{E_v} \boxed{\sum_{h_1}\sum_{h_2}\dots\sum_{h_n} e^{h_1\Delta_1} e^{h_2\Delta_2} \dots e^{h_n\Delta_n}}$$

$$= e^{E_v} \sum_{h_1} e^{h_1\Delta_1} \sum_{h_2} e^{h_2\Delta_2} \dots \sum_{h_n} e^{h_n\Delta_n} =$$

Finding expression for p(h| v)

Due to the structure of RBMs the conditional probability p(h| v) can be found analytically

$$-\sum_{\boldsymbol{h}} \boxed{p(\boldsymbol{h} \mid \boldsymbol{v})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

Let's group E by h_i

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i}^{n} h_i \boxed{\left(\sum_{j}^{m} \omega_{ij} v_j + c_i\right)} - \boxed{\sum_{j}^{m} b_j v_j}$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i}^{n} h_i \Delta_i - E_v$$

$$p(\boldsymbol{h} \mid \boldsymbol{v}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{p(\boldsymbol{v})} = \frac{\frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}$$

Let's write it explicitly

$$e^{-E(\mathbf{v},\mathbf{h})} = e^{E_v} \sum_{h_1} e^{h_1\Delta_1} \sum_{h_2} e^{h_2\Delta_2} \ldots \boxed{\sum_{h_n} e^{h_n\Delta_n}} =$$

h_i is a binary variable: $h_i = \{0, 1\}$

$$\sum_{h_i} e^{h_i\Delta_i} = 1 + e^{\Delta_i}$$

This gives

$$\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} = e^{E_v}\left(1+e^{\Delta_1}\right)\ldots\left(1+e^{\Delta_n}\right) = e^{E_v}\prod_{i}^{n}\left(1+e^{\Delta_i}\right)$$

Finding expression for p(h| v)

Due to the structure of RBMs the conditional probability p(h| v) can be found analytically

$$-\sum_{\boldsymbol{h}} \boxed{p(\boldsymbol{h}\,|\,\boldsymbol{v})} \frac{\partial E(\boldsymbol{v},\boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

Let's group E by h_i

$$E(\mathbf{v},\mathbf{h}) = -\sum_i^n h_i \boxed{\left(\sum_j^m \omega_{ij} v_j + c_i\right)} - \boxed{\sum_j^m b_j v_j}$$

$$E(\mathbf{v},\mathbf{h}) = -\sum_i^n h_i \Delta_i - E_v$$

$$p(\boldsymbol{h}\,|\,\boldsymbol{v}) = \frac{p(\boldsymbol{v},\boldsymbol{h})}{p(\boldsymbol{v})} = \frac{\frac{1}{Z} e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}}$$

Finally we have

$$\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} = e^{E_v} \prod_i^n \left(1 + e^{\Delta_i}\right) \qquad e^{-E(\mathbf{v},\mathbf{h})} = e^{E_v} \prod_i^n e^{h_i \Delta_i}$$

From this we can compute conditional probability

$$p(\mathbf{h}|\mathbf{v}) = \frac{e^{E_v} \prod_i^n e^{h_i \Delta_i}}{e^{E_v} \prod_i^n \left(1 + e^{\Delta_i}\right)} = \prod_i^n \boxed{\frac{e^{h_i \Delta_i}}{1 + e^{\Delta_i}}}$$

$$p(\mathbf{h}|\mathbf{v}) = \prod_i^n \boxed{p(h_i|\mathbf{v})}$$

Probability of activation of i-th hidden unit given visible units **v**

$$p(h_i = 1|\mathbf{v}) = \frac{e^{\Delta_i}}{1 + e^{\Delta_i}} = \frac{1}{1 + e^{-\Delta_i}} = \operatorname{sig}(\Delta_i)$$

Finding expression for p(h| v)

Due to the structure of RBMs the conditional probability p(h| v) can be found analytically

$$-\sum_{\boldsymbol{h}} \boxed{p(\boldsymbol{h} \,|\, \boldsymbol{v})} \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial \boldsymbol{\theta}}$$

$$p(\mathbf{h}|\mathbf{v}) = \prod_i^n p(h_i|\mathbf{v})$$

$$\begin{aligned} p(h_i = 1|\mathbf{v}) &= \mathrm{sig}\,(+\Delta_\mathrm{i}) \\ p(h_i = 0|\mathbf{v}) &= \mathrm{sig}\,(-\Delta_\mathrm{i}) \end{aligned}$$

$$\boxed{\Delta_i = \sum_j^m \omega_{ij} v_j + c_i}$$

Conclusions:

- *RBM works as stochastic neuron with sigmoid activation function*
- *Each hidden unit is independent (restriction in hidden-hidden connections)*
- *Similarly for visible units*

$$p(H_i = 1 \,|\, \boldsymbol{v}) = \mathrm{sig}\left(\sum_{j=1}^m w_{ij} v_j + c_i\right)$$

$$p(V_j = 1 \,|\, \boldsymbol{h}) = \mathrm{sig}\left(\sum_{i=1}^n w_{ij} h_i + b_j\right)$$

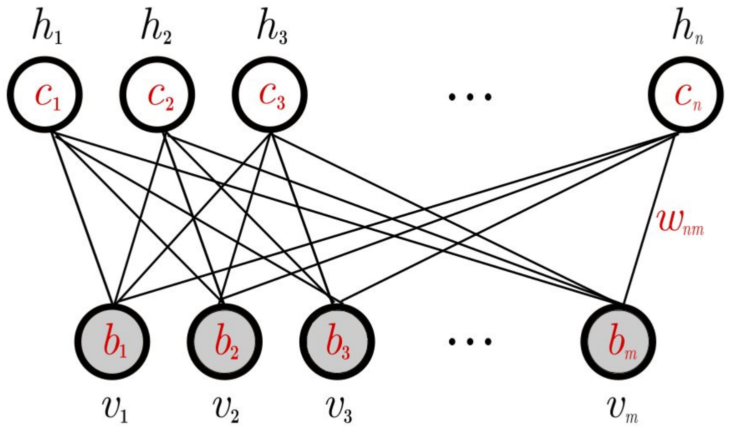


Fig. 5. The network graph of an RBM with n hidden and m visible units.

The gradient of L

$$p(h_i = 1|\mathbf{v}) = \text{sig}\left(+\Delta_\text{i}\right)$$
$$p(h_i = 0|\mathbf{v}) = \text{sig}\left(-\Delta_\text{i}\right)$$
$$\Delta_i = \sum_{j}^{m} \omega_{ij} v_j + c_i$$

The gradient can be computed using similar factorization method

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

After few steps of calculations we finally obtain

$$\sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}} = \text{sig}\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right) v_j$$

The final expression for the gradient of log-likelihood is given:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}} + \sum_{\boldsymbol{v}, \boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{v}, \boldsymbol{h})}{\partial w_{ij}}$$

This is still intractable

$$= \sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) h_i v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) \sum_{\boldsymbol{h}} p(\boldsymbol{h} \,|\, \boldsymbol{v}) h_i v_j = p(H_i = 1 |\, \boldsymbol{v}) v_j - \boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 |\, \boldsymbol{v}) v_j}$$

The gradient of L (per one example)

Final expressions:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}} = p(H_i = 1 | \boldsymbol{v}) v_j - \boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 | \boldsymbol{v}) v_j}$$

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial b_j} = v_j - \boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j}$$

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial c_i} = p(H_i = 1 | \boldsymbol{v}) - \boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 | \boldsymbol{v})}$$

probability of firing rate of the neuron

$$p(H_i = 1 \,|\, \boldsymbol{v}) = \mathrm{sig}\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right)$$

marginal probability

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}$$

probability of **v** generated by model

Minibatch - what is at equilibrium?

For **mini-batch** algorithm one computes $\frac{1}{\ell} \sum_{\boldsymbol{v} \in S} \frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}}$

Equilibrium condition: *consider l = infinity, and gradient zero*

$$\frac{1}{l} \sum_{\mathbf{v} \in \mathcal{S}} \frac{\partial \ln \mathcal{L}(\theta | \mathbf{v})}{\partial \omega_{ij}} = \frac{1}{l} \sum_{\mathbf{v} \in \mathcal{S}} p(h_i = 1 | \mathbf{v}) v_j - \sum_{\mathbf{v}'} p(\mathbf{v}') p(h_i = 1 | \mathbf{v}') v'_j = \sum_{\mathbf{v}} q(\mathbf{v}) p(h_i = 1 | \mathbf{v}) v_j - \sum_{\mathbf{v}'} p(\mathbf{v}') p(h_i = 1 | \mathbf{v}') v'_j = 0$$

$$\sum_{\mathbf{v}} p(h_i = 1 | \mathbf{v}) v_j (q(\mathbf{v}) - p(\mathbf{v})) = 0 \Longrightarrow \boxed{q(\mathbf{v}) = p(\mathbf{v})}$$

for large mini-batch model and data distributions should be the same

Approximating gradients by sampling

The gradient with respect to **b** vector

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial b_j} = v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j$$

expectation value of **v**
over the model probability

We do not know nice and
compact form of p(**v**) :(

Let's us recall simple example of sampling from gaussian distribution **rho(x)**

$$\mathbb{E}_{\rho(x)}\left[f(x)\right] = \int_{-\infty}^{+\infty} \rho(x) f(x) dx$$

This integral can be approximated by average of sampled x from rho distribution

$$\mathbb{E}_{\rho(x)}\left[f(x)\right] \approx \frac{1}{N} \sum_{x_i \sim \rho(x)} f(x_i)$$

x is sampled from rho(x)

A toy example of sampling approach

Consider following example:

$$\rho(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \qquad f(x) = x^2$$

$$\mathbb{E}_{\rho(x)}\left[f(x)\right] = \int_{-\infty}^{+\infty} \rho(x) f(x) dx = 1$$

Approximation by sampling x from rho

$$\mathbb{E}_{\rho(x)}\left[f(x)\right] \approx \frac{1}{N} \sum_{x_i \sim \rho(x)} f(x_i)$$

```
for n = 1:6
  E = mean((randn(10^n).^2))
  println("$(10^n)\t$E")
end
```

N	E(x^2)
10	0.351830
100	1.046075
1000	0.939235
10000	1.027309
100000	1.001297
1000000	0.999072

In the example above we have sampled x from rho, however in real world examples we don't know exact rho or we cannot easily sample from rho ;(

Approximating gradients by sampling

The gradient with respect to **b** vector $\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial b_j} = v_j - \boxed{\sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j}$

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial w_{ij}} = p(H_i = 1 | \, \boldsymbol{v}) v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 | \, \boldsymbol{v}) v_j$$

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial b_j} = v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j$$

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial c_i} = p(H_i = 1 | \, \boldsymbol{v}) - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) p(H_i = 1 | \, \boldsymbol{v})$$

Let's come back to the RBMs case:

$$\sum_{\mathbf{v}} p(\mathbf{v}) f(\mathbf{v}) = \mathbb{E}_{p(\mathbf{v})} \left[f(\mathbf{v}) \right] \approx \frac{1}{N} \sum_{\mathbf{v} \sim p(\mathbf{v})} f(\mathbf{v})$$

N is a number of samples generated by p(**v**)

We want to approximate expectation value with sampling.
The only question which left is:

How to sample from complex distribution p(**v**)?

Markov chain Monte Carlo method (MCMCM)

We want to approximate:

$$\sum_{\mathbf{v}} p(\mathbf{v}) f(\mathbf{v}) = \mathbb{E}_{p(\mathbf{v})}\left[f(\mathbf{v})\right] \approx \frac{1}{N} \sum_{\mathbf{v} \sim p(\mathbf{v})} f(\mathbf{v})$$

This can be done via sampling method

Some definitions:

- **Markov chain** is a time discrete stochastic process, where the next state of the system depends only on current state of the system

memoryless process

$$p_{ij}^{(k)} = \Pr\left(X^{(k+1)} = j \,|\, X^{(k)} = i, X^{(k-1)} = i_{k-1}, \ldots, X^{(0)} = i_0\right) = \Pr\left(X^{(k+1)} = j \,|\, X^{(k)} = i\right)$$

next state current state

- Many more: irreducible, aperiodic, detailed balance condition…. to define when we can use MCMC methods (stationary solution)

Markov chain can be generated with Metropolis sampling however this has some drawbacks:
states can be rejected :(

Solution? Gibbs sampling method - no rejections!

Gibbs sampling method for RBMs

We want to approximate:

$$\sum_{\mathbf{v}} p(\mathbf{v}) f(\mathbf{v}) = \mathbb{E}_{p(\mathbf{v})}\left[f(\mathbf{v})\right] \approx \frac{1}{N} \sum_{\mathbf{v} \sim p(\mathbf{v})} f(\mathbf{v})$$

This can be done via Gibbs sampling method:

- if the target distribution is p(**x**) we must have analytical expressions for conditional probabilities: $p(x_i | x_j)$ for j<>i

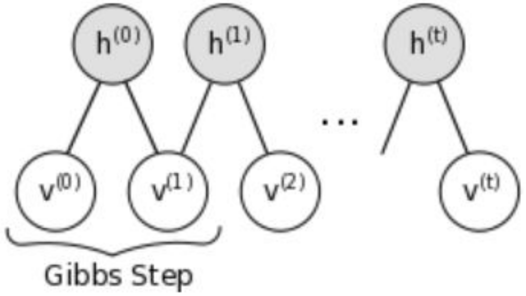
$$p(H_i = 1 \,|\, \boldsymbol{v}) = \mathrm{sig}\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right)$$

Done!

$$p(V_j = 1 \,|\, \boldsymbol{h}) = \mathrm{sig}\left(\sum_{i=1}^{n} w_{ij} h_i + b_j\right)$$

- we must be able to sample from those probabilities e.g if p(h=1|v) = 0.9 > rand() => h=1 else h=0 **Done!**

The algorithm:



$$h^{(n+1)} \sim sigm(W' v^{(n)} + c)$$
$$v^{(n+1)} \sim sigm(W h^{(n+1)} + b),$$

after some steps $\mathbf{v}^k$ is a sample from p(**v**) distribution.

Gibbs sampling method for RBMs

We want to approximate:

$$\sum_{\mathbf{v}} p(\mathbf{v}) f(\mathbf{v}) = \mathbb{E}_{p(\mathbf{v})}\left[f(\mathbf{v})\right] \approx \frac{1}{N} \sum_{\mathbf{v} \sim p(\mathbf{v})} f(\mathbf{v})$$

Let's back to RBM we can compute p(**v**) using exact approach

```
rbm = RBM(9, 9)
vp = get_permutations(length(rbm.bv))
hp = get_permutations(length(rbm.bh))
Z = Z_exact(rbm, hp, vp)  1.13e+6…
```

```
probs_pv = zeros(length(vp))
for s = 1:length(vp)
  probs_pv[s] = p_v_exact(vp[s], Z, hp, rbm)
end ✔
```

p_v_exact:

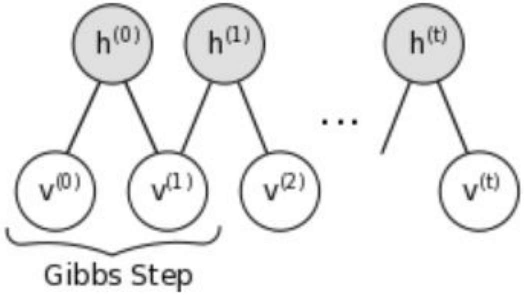
$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}$$

Let's sample p(v) with Gibbs method

```
vk = copy(samples[1,:]) # initialize with data sample
gibbs_pv = zeros(nvp)
k  = 100000
for i = 1:k
  hk = sample_bernoulli(p_h_cond_v(vk, rbm))
  vk = sample_bernoulli(p_v_cond_h(hk, rbm))
  idx = parse(Int64, join(vk), 2) + 1 # get idx
  gibbs_pv[idx] += 1.0/k
end ✔
```

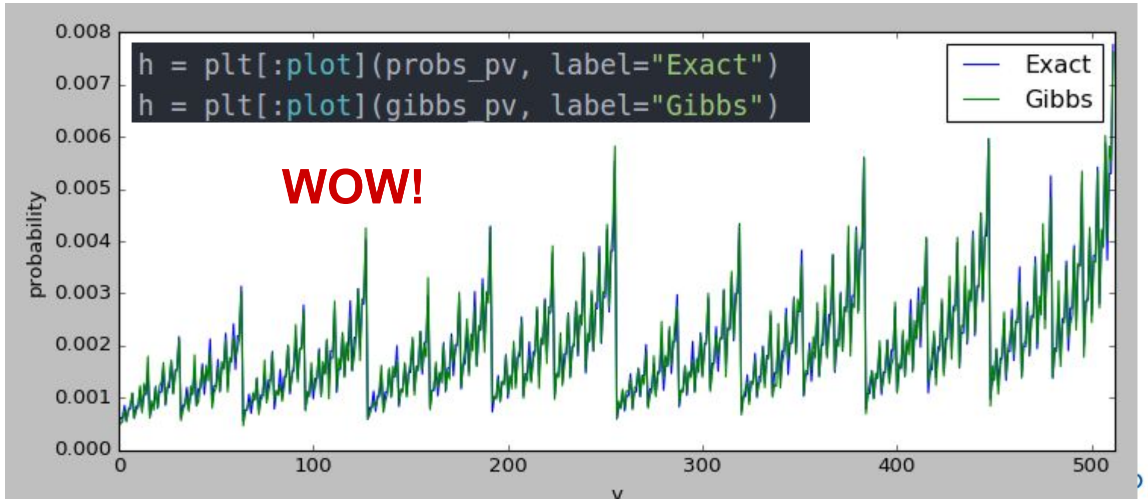
```
function sample_bernoulli(probs)
  return 1(rand(length(probs)) .< probs)
end
```

Gibbs sampling



$$p(H_i = 1 \mid \boldsymbol{v}) = \mathrm{sig}\left(\sum_{j=1}^{m} w_{ij} v_j + c_i\right)$$

$$p(V_j = 1 \mid \boldsymbol{h}) = \mathrm{sig}\left(\sum_{i=1}^{n} w_{ij} h_i + b_j\right)$$



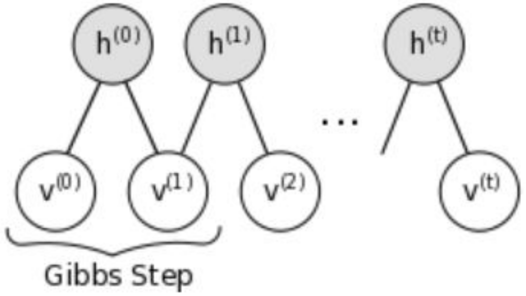
Gibbs sampling method for RBMs

We want to approximate:

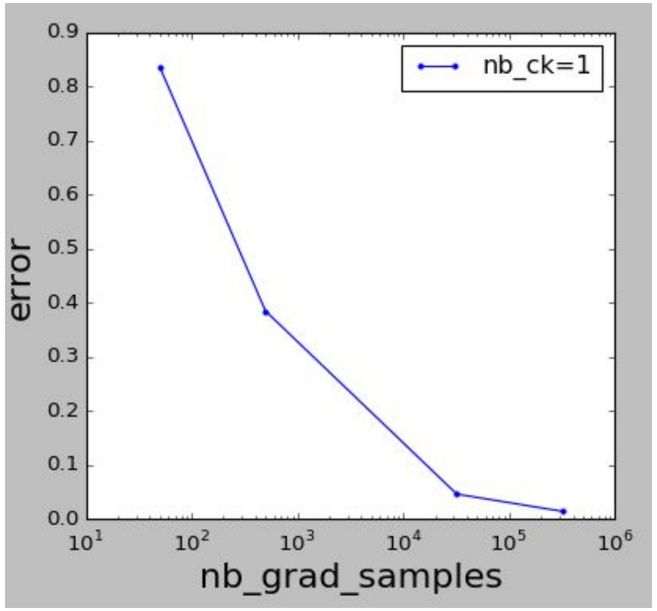
$$\sum_{\mathbf{v}} p(\mathbf{v}) f(\mathbf{v}) = \mathbb{E}_{p(\mathbf{v})}\left[f(\mathbf{v})\right] \approx \frac{1}{N} \sum_{\mathbf{v} \sim p(\mathbf{v})} f(\mathbf{v})$$

Gibbs sampling allows us to sample $\mathbf{v} \sim p(\mathbf{v})$

Gibbs sampling



How many samples k should we compute to estimate gradients properly?



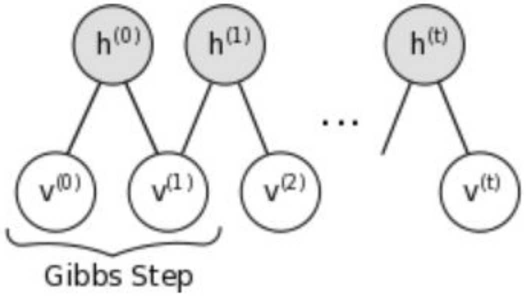
Relative error of gradient computation
using Gibbs sampling and exact

What people do?

They use k = 1

Contrastive divergence

Gibbs sampling



The idea of k-step contrastive divergence learning (CD-k) is quite simple: instead of approximating the second term in the log-likelihood gradient by a sample from the RBM-distribution (which would require running a Markov chain until the stationary distribution is reached), a Gibbs chain is run for only k steps (and usually $k = 1$). The Gibbs chain is initialized with a training example $\boldsymbol{v}^{(0)}$ of the training set and yields the sample $\boldsymbol{v}^{(k)}$ after k steps. Each step t consists of sampling $\boldsymbol{h}^{(t)}$ from

Contrastive divergence

Algorithm 1: k-step contrastive divergence

Input: RBM $(V_1, \dots, V_m, H_1, \dots, H_n)$, training batch S
Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i = 1, \dots, n$, $j = 1, \dots, m$

init $\Delta w_{ij} = \Delta b_j = \Delta c_i = 0$ for $i = 1, \dots, n$, $j = 1, \dots, m$
forall the $\boldsymbol{v} \in S$ **do**
- $\boldsymbol{v}^{(0)} \leftarrow \boldsymbol{v}$
- **for** $t = 0, \dots, k-1$ **do**
  - **for** $i = 1, \dots, n$ **do** sample $h_i^{(t)} \sim p(h_i \,|\, \boldsymbol{v}^{(t)})$
  - ;
  - **for** $j = 1, \dots, m$ **do** sample $v_j^{(t+1)} \sim p(v_j \,|\, \boldsymbol{h}^{(t)})$
  - ;
- **for** $i = 1, \dots, n$, $j = 1, \dots, m$ **do**
  - $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i = 1 \,|\, \boldsymbol{v}^{(0)}) \cdot v_j^{(0)} - p(H_i = 1 \,|\, \boldsymbol{v}^{(k)}) \cdot v_j^{(k)}$
- **for** $j = 1, \dots, m$ **do**
  - $\Delta b_j \leftarrow \Delta b_j + v_j^{(0)} - v_j^{(k)}$
- **for** $i = 1, \dots, n$ **do**
  - $\Delta c_i \leftarrow \Delta c_i + p(H_i = 1 \,|\, \boldsymbol{v}^{(0)}) - p(H_i = 1 \,|\, \boldsymbol{v}^{(k)})$

This seems to work quite well but for small values of k one has to be aware of bad approximation of gradients - gradient is biased (see more in refs).

$$\boxed{\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \boldsymbol{v})}{\partial b_j} = v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v}) v_j}$$

Persistent contrastive divergence (PCD)

Algorithm 1: k-step contrastive divergence

Input: RBM $(V_1, \ldots, V_m, H_1, \ldots, H_n)$, training batch S

Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i = 1, \ldots, n$, $j = 1, \ldots, m$

```
init Δw_ij = Δb_j = Δc_i = 0 for i = 1,...,n, j = 1,...,m
forall the v ∈ S do
    v^(0) ← v
    for t = 0,...,k−1 do
        for i = 1,...,n do  sample h_i^(t) ~ p(h_i | v^(t))
        ;
        for j = 1,...,m do  sample v_j^(t+1) ~ p(v_j | h^(t))
        ;
    for i = 1,...,n, j = 1,...,m do
        Δw_ij ← Δw_ij + p(H_i = 1 | v^(0))· v_j^(0) − p(H_i = 1 | v^(k))· v_j^(k)
    for j = 1,...,m do
        Δb_j ← Δb_j + v_j^(0) − v_j^(k)
    for i = 1,...,n do
        Δc_i ← Δc_i + p(H_i = 1 | v^(0)) − p(H_i = 1 | v^(k))
```

don't re-initialize $\mathbf{v}^{(0)}$ for each example in mini-batch

Dependence on k

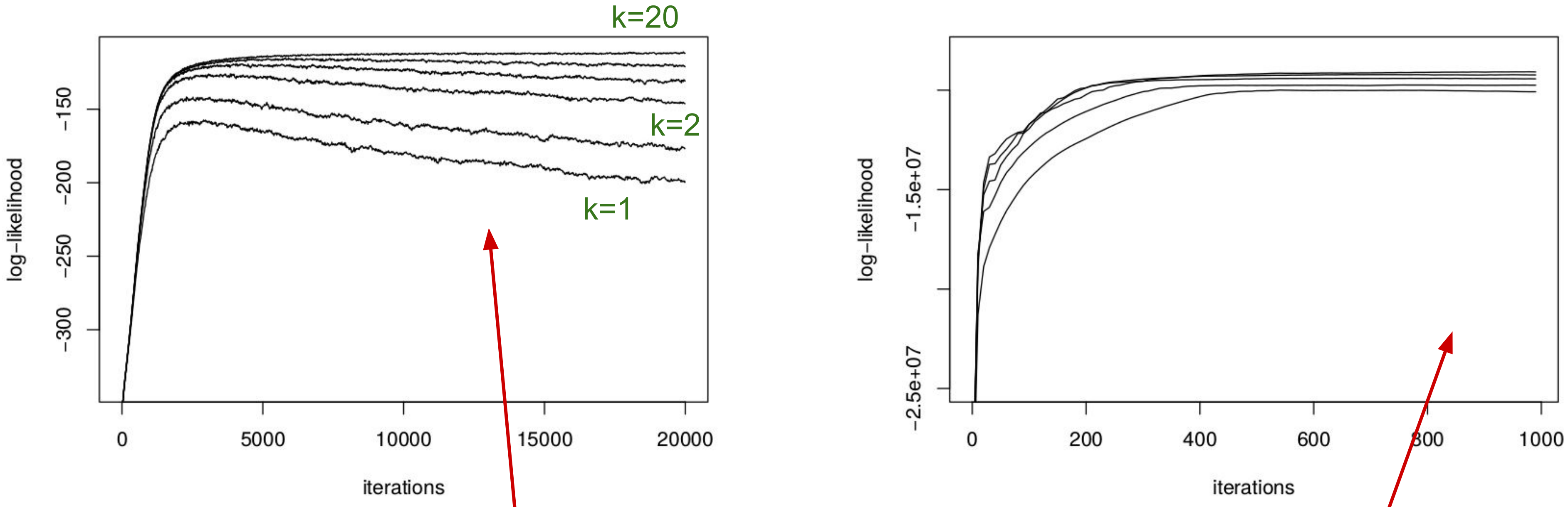


Fig. 9. Evolution of the log-likelihood during training of RBMs with CD-k where different values for k were used. The left plot shows the results for BAS (from bottom to top $k = 1, 2, 5, 10, 20, 100$) and the right plot for MNIST (from bottom to top $k = 1, 2, 5, 10, 20$). The values are medians over 25 runs.

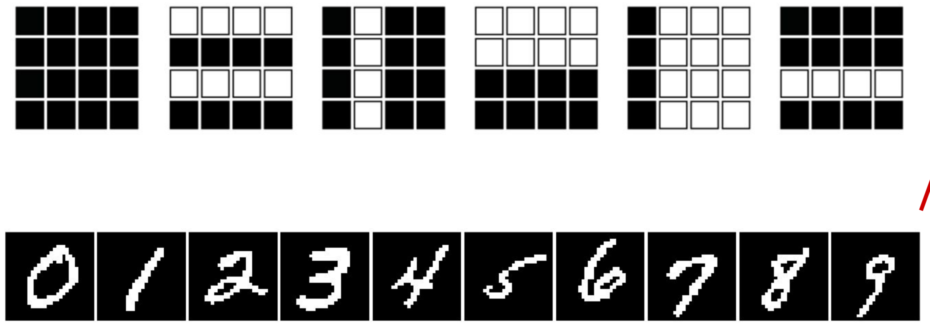


Fig. 6. Top: Patterns from the BAS data set. Bottom: Images from the MNIST data set.

Summary

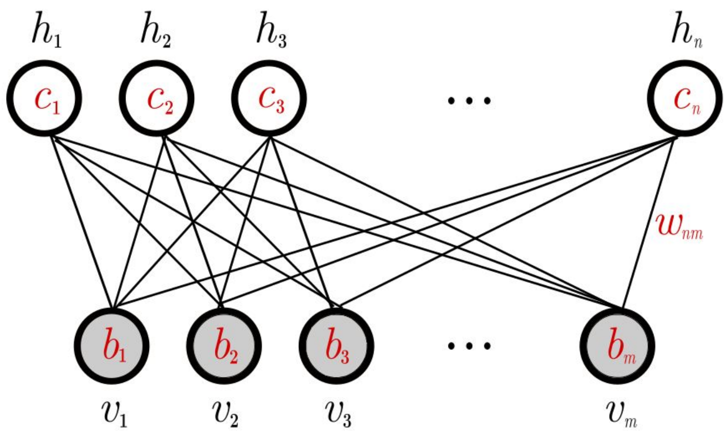


Fig. 5. The network graph of an RBM with n hidden and m visible units.

RBM is an energy based model

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_iv_j - \sum_{j=1}^{m} b_jv_j - \sum_{i=1}^{n} c_ih_i$$

We want to maximize log likelihood of p(v)

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

Using gradient ascent and CD approximation method

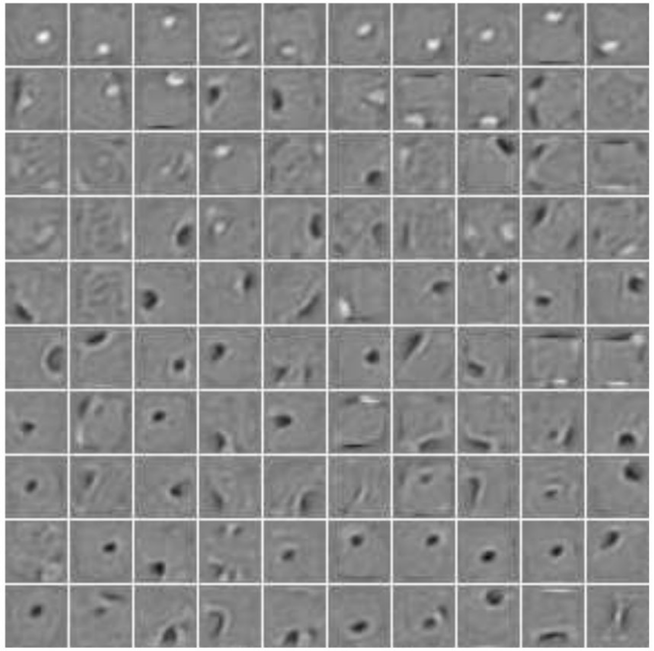
$$\frac{\partial \ln\mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{v})}{\partial w_{ij}} = p(H_i = 1|\,\boldsymbol{v})v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i = 1|\,\boldsymbol{v})v_j$$

$$\frac{\partial \ln\mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{v})}{\partial b_j} = v_j - \sum_{\boldsymbol{v}} p(\boldsymbol{v})v_j$$

$$\frac{\partial \ln\mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{v})}{\partial c_i} = p(H_i = 1|\,\boldsymbol{v}) - \sum_{\boldsymbol{v}} p(\boldsymbol{v})p(H_i = 1|\,\boldsymbol{v})$$

Applications of RBMs

- RBMs can be used to pretrain weights for supervised learning

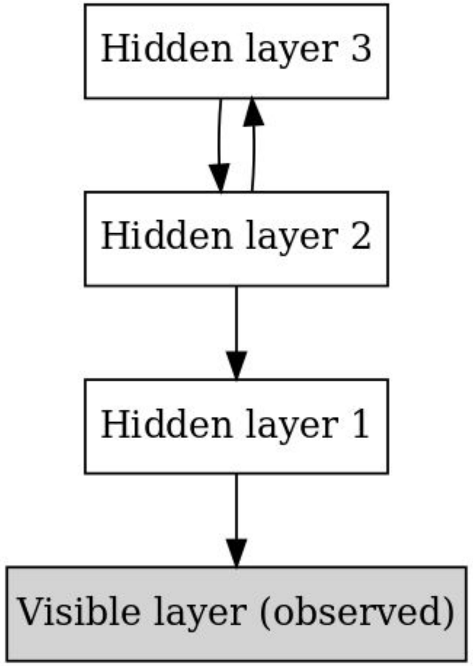


- RBMs are generative model, one can solve in-painting problem with them

$$p(V_{o+1}, \dots, V_m \mid V_1 = v_1, \dots, V_o = v_o)$$

keep selected units fixed and sample rest using Gibbs sampling

- Train Deep Belief networks
- RBMs were used for recommendations :)



Notes: Why don't we work with Energy?

- RBM is an energy based model, why not minimize energy E(v) instead of p(v)

$$E(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

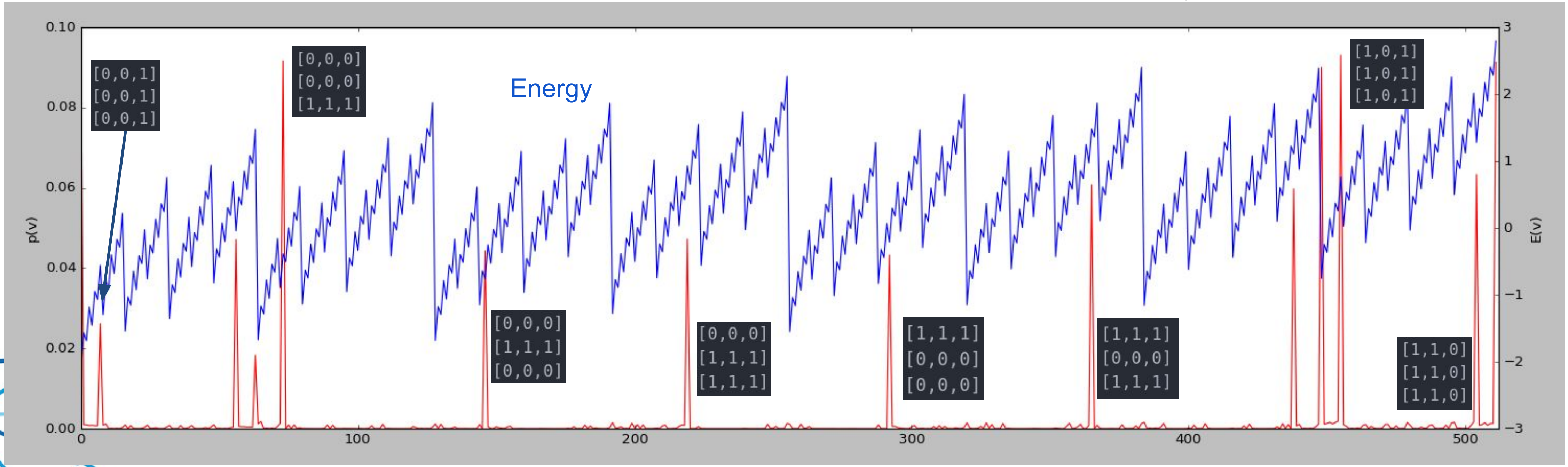
$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

$$E(\mathbf{v}) = \frac{1}{N_h}\sum_{\mathbf{h}} E(\mathbf{v},\mathbf{h})$$

Smaller energy => higher probability

- This is not true in general for E(**v**)
- Energy is not bounded, probability must be normalize

Results for *converged* 3x3 BAS dataset



Conclusions

Don’t use RBMs if you don’t have to

References

- https://theclevermachine.wordpress.com/page/3/ -mcmc
- https://theclevermachine.wordpress.com/page/2/ gibbs sampling
- http://deeplearning.net/tutorial/rbm.html - theano tutorial on RBMs (full implementation)
- Training RBMs: An introduction - the best description on how do the RBMs work

FORNAX

WWW.FORNAX.CO